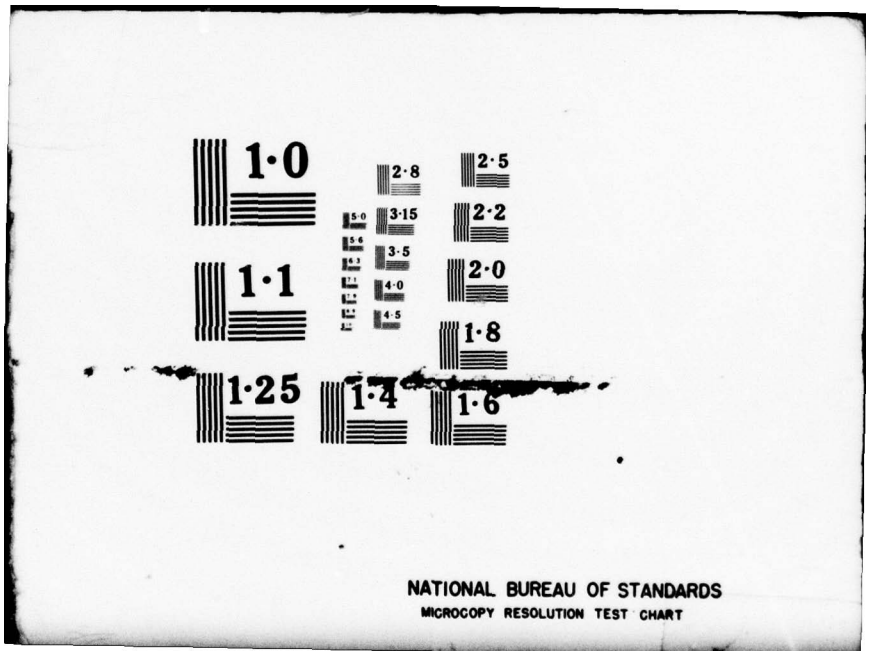
AD-A055 975 NAVAL UNDERWATER SYSTEMS CENTER NEW LONDON CONN NEW --ETC F/G 9/2
A COMPARISON OF TACTICAL MILITARY AND COMMERCIAL DATA PROCESSIN--ETC(U)
MAY 78 L A DENOIA, R L GORDON

UNCLASSIFIED NUSC-TR-5783 NL

1 OF 1
ADA
055975

END
DATE
FILMED
8-78
DDC


1·0
1·1
1·25
1·4
1·6
1·8
2·0
2·2
2·5
2·8
3·15
3·5
4·0
4·5
5·0
NATIONAL BUREAU OF STANDARDS
MICROCOPY RESOLUTION TEST CHART

AD A055975
NUSC Technical Report 5783
AD No.
DDC FILE COPY

NUSC Technical Report 5783

LEVEL II

A Comparison of Tactical Military and Commercial Data Processing Computer Architectural Requirements

Lynn A. DeNoia
Robert L. Gordon
Special Projects Department

DDC
JUL 5 1978
F

23 May 1978

NUSC
NAVAL UNDERWATER SYSTEMS CENTER
Newport, Rhode Island • New London, Connecticut

Approved for public release; distribution unlimited.

78 06 27 108

PREFACE

This report was prepared under NUSC Project No. A-750-26, "AADC Data Processing Requirements." The sponsoring activity is the Naval Research Laboratory, W. Smith, Program Manager.

The Technical Reviewer was P. R. Bono (Code 3142).

REVIEWED AND APPROVED: 23 May 1978

R. W. Hasse
Head: Special Projects Department

The authors of this report are located at the New London Laboratory, Naval Underwater Systems Center, New London, Connecticut 06320.

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM	
1. REPORT NUMBER: TR 5783 NUSC-TR-5783	2. GOVT ACCESSION NO. / 3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): A COMPARISON OF TACTICAL MILITARY AND COMMERCIAL DATA PROCESSING COMPUTER ARCHITECTURAL REQUIREMENTS	5. TYPE OF REPORT & PERIOD COVERED: Technical Rept. / 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Lynn A. DeNoia, Robert L. Gordon	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Naval Underwater Systems Center, New London Laboratory, New London, Connecticut 06320	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: A-750-26
11. CONTROLLING OFFICE NAME AND ADDRESS: Naval Research Laboratories (Code 5400), Washington, DC 20375	12. REPORT DATE: 23 May 1978 / 13. NUMBER OF PAGES: 20 (12) 34 P.
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): UNCLASSIFIED / 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

DDC JUL 5 1978 RECEIVED F

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Commercial Data Processing
Computer-Family Architecture
Cost/Performance Requirements
Input/Output Architecture
Interface Control
Standardization
Tactical Military Data Processing

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

It is shown that tactical military data processing need not be different from commercial data processing in terms of the computer architectural features required to support applications programs. The general lack of specific performance data from execution of operational tactical programs makes it difficult to evaluate architectural issues with current military computers or to compare future candidates; however, sufficient data are available to develop general conclusions concerning computer architectures.

(over)

DD FORM 1 JAN 73 1473

405 918 78 06 27 109 ke

20. (Cont'd)

It is recommended that the military capitalize on currently available commercial computer architectures and keep careful track of commercial research and development efforts.

TABLE OF CONTENTS

	Page
LIST OF ILLUSTRATIONS	ii
LIST OF TABLES	ii
INTRODUCTION	1
Motivation	1
Levels of Standardization	1
Related Architectural Issues	3
HOW TO CHARACTERIZE AN APPLICATION	3
Top-Down Approach	4
Program Statistics	7
Architectural Impact of Application Data	10
CHARACTERIZING TACTICAL MILITARY DATA PROCESSING	11
Data Processing Versus Signal Processing	11
Functional Characterization	11
ANALOGOUS COMMERCIAL DATA PROCESSING CHARACTERISTICS	17
COMMERCIAL MACHINES IN TACTICAL USE	19
A BRIEF HISTORY OF MILITARY DATA PROCESSING	19
CURRENT AND FUTURE REQUIREMENTS	22
RECOMMENDATIONS	23
SUMMARY	24
REFERENCES	25
BIBLIOGRAPHY	27

ACCESSION for
NTIS
DDC
UNANNOUNCED
JUSTIFICATION
BY
DISTRIBUTION/AVAILABILITY CODES
Dist. AVAIL. and/or SPECIAL
A1

RE: Classified references, NUSC-TR-5783
Document should remain for unlimited distribution per Ms. D'Andelis, NUSC

LIST OF ILLUSTRATIONS

Figure		Page
1	Interdependencies of Application Functions	5

LIST OF TABLES

Table		Page
1	Some Data Processing Requirement Details Having Specific Computer Architectural Implications	4
2	Architecture Parameters	8
3	Measurable Architectural Characteristics	8
4	Source Code Summary for a Tactical Data Processing System (AN/UYK-7 CPU)	12
5	Execution Summary for a Tactical Data Processing System (USQ-20 Computer)	14
6	Summary of the Results from a Commercial System Study	17
7	Future Navy Computer Architectural Requirements	22
8	Future Air Force Computer Functional Requirements	23

A COMPARISON OF TACTICAL MILITARY AND COMMERCIAL DATA PROCESSING COMPUTER ARCHITECTURAL REQUIREMENTS

INTRODUCTION

MOTIVATION

The proliferation of different types of computers within the Department of Defense (DOD) at all levels has recently become a source of general concern. Such proliferation is a major factor both in the high costs of developing and maintaining military software and in the problems of reliability, training, logistics, and hardware maintenance. The Navy/Army Computer Family Architecture (CFA) program[1] has addressed this situation by selecting an existing computer architecture[2] that can serve as the basis for a standard military family of computers. The family concept allows presentation to the programmers of a uniform set of upward-compatible (in software) machines that can be implemented across a spectrum of hardware technology to produce individual machines tailored to particular cost/performance requirements. This means stabilizing software at the computer architectural level. Such stabilization has been commercially practiced for at least a decade, allowing a cost/performance range of 500:1 to be implemented by means of a rapidly changing technology. Standardization on a single family of computers allows for portable software, universally applicable training, reduced maintenance costs, and simplified information exchange while still encouraging a wide variety of implementations to cover the cost/performance spectrum of all military applications. Benefits could apply not only within the Navy (among various platforms, for example) but also interservice- and intersystem-wide within DOD.

Intraservice compatibility among platforms and interservice compatibility of information are already important goals that represent the key to providing increasing capabilities to share both information and processing loads in appropriate situations. By promoting compatibilities at the computer architectural level, we note that management of higher level requirements (e.g., data systems, networks, etc.) becomes more reasonable.

LEVELS OF STANDARDIZATION

It is important to distinguish clearly between (1) computer architecture as it is the concern of CFA and this report and (2) the physical details of how that architecture is implemented. The architecture of a

machine is the logical description of what it can do--exactly and completely what the programmer needs to know in order to write software. It is usually presented in a document called the "Principles of Operation" (P of O). The architecture or logical structure shall be kept totally distinct from the implementation where implementation includes details such as what technology is applied, how the registers are built and where they are located, or in what range of environmental conditions the equipment will operate properly.

Notice that it is also important to draw a boundary between two components of computer architecture, the instruction set or central processor architecture and the input/output (I/O) architecture. The latter necessarily might be highly hardware- or implementation-dependent. The interface between these two can be clearly defined in a few instructions (such as start and halt I/O, the test for status) belonging to the Central Processing Unit (CPU) instruction set architecture, while leaving the details to a separate I/O or so-called "channel" architecture.

For well-designed computer systems, there may be other levels of compatibility. Standardization at an operating system level is desirable to promote the sharing and stabilization of data processing application programs and to allow a single set of program development tools and support software to be applied to and amortized over all models of the baseline architecture. It implies compatibility both at the operating system interface and for nonprivileged (problem state) instructions. Instructions that must operate in some type of privileged mode may not need to be compatible (changes can be masked by small changes in the operating system) if a great deal of care is put into standardizing the operating system interfaces. This separation of user mode from privileged or operating system mode is crucial for identifying the set of programs or routines that can be used across the family of available computers, and is required if computer software systems are to be protected against accidental or intentional software faults.

It is interesting to consider the difference in defining an architecture (interface) at the instruction set level from one at the higher order language (HOL) level. In principle, both are possible, although experience and technical evaluations* have shown that it is much more difficult to specify a HOL interface (e.g., incompatibilities of so-called standard Fortrans and the Navy's Cobol standardization efforts). Computer manufacturers have generally decided to standardize at the instruction set level. Even so, most P's of O are not complete enough to enable programmers to write software without resorting to test cases on a real machine for individual instruction sequences.

*Lack of semantic completeness is even greater in HOL descriptions, where the individual functions are more powerful.

RELATED ARCHITECTURAL ISSUES

With these ideas in mind, the goal of this report is to establish what are the military tactical data processing computer architectural requirements. Armed with such information, we expect to be able to establish confidence in the ability of any particular architecture to meet tactical data processing needs. Our approach has been to survey the literature, to examine available tactical systems in detail, and to listen to as many people as possible who are responsible for planning and implementing tactical data processing systems. We expect some of our conclusions will be controversial, and we look forward to what informed debate and careful system scrutiny this should promote. DOD can well benefit from intelligent detailed examination of real tactical military data processing requirements and practices upon which improvements can be based.

Trends in data processing are indicating an ever-narrowing gap between military and commercial applications.[3,4] This will be discussed later in more detail--it is presented now to pique your curiosity and to serve as food for thought as you peruse the next sections.

An issue that must be carefully considered for military and commercial implications is the effect standardization is beginning to have on the computer industry. New interfaces are being defined by major manufacturers so that standard functional modules can be put together with relatively little difficulty, alleviating the kinds of problems facing DOD. Included are various levels of standard hardware modules and standard software modules that need not be continually redesigned and/or programmed. If care is taken in such interface definitions, new technologies will be easy to fit into such kinds of systems. Notice that the current emphasis is on interface control, and recognize that standardization in general must be founded on definitions that will be flexible enough to not only allow, but actually promote, the introduction of new technology. Instruction set standardization was discussed above as an example of one such interface. Data communications is another area with lots of current activity in interface specification. IBM's System Network Architecture (SNA) and Digital Equipment Corporation's DECnet are examples of standards that have been proposed. The military must remain keenly aware of developing commercial standards and user acceptance levels so that their own specifications can be both compatible and flexible.

HOW TO CHARACTERIZE AN APPLICATION

We are primarily concerned with those ways to characterize the data processing requirements that have a major impact on the architectural features of a computer. It is first necessary to examine applications

for both generic and specific requirements that can be used either (1) to characterize the specific processing requirements of a particular application or (2) to imply necessary computer capabilities for similar applications.

TOP-DOWN APPROACH

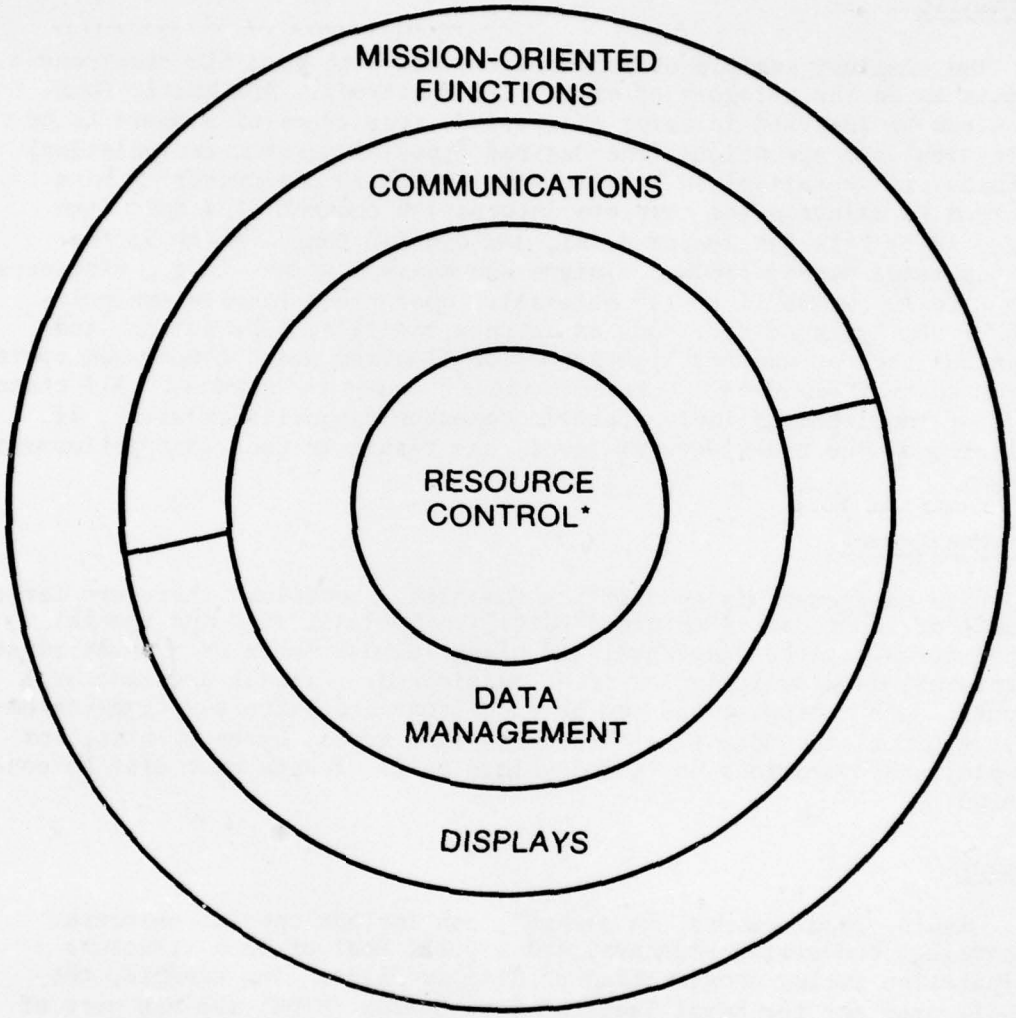
A top-down approach to the characterization of applications begins with a survey of some generic functional categories that influence computer architectures, such as

- arithmetic
- display support
- communication
- data management
- resource control.

If we use the word "resource" in a general sense, these categories may be arranged in a hierarchical format to help identify their interdependencies (figure 1). This is meant as a conceptual aid, not as a definition of any strict boundaries. Successive refinement of these general functional categories, however, should lead to levels where requirement details can be analyzed for specific computer architectural implications such as those listed in table 1.

Table 1. Some Data Processing Requirement Details Having Specific Computer Architectural Implications

- language linkages and conventions
- process communication requirements
- event/interrupt handling and processing
- data type usage
- data structures used
- data conversions needed
- locality of reference of information and data
- arithmetic functions needed
- special character processing or editing requirements
- resource management and control requirements
- fail-safe requirements, reliability, and error recovery



*ARITHMETIC IS AN ATTRIBUTE OF THE CENTRAL PROCESSING UNIT RESOURCE.

Figure 1. Interdependencies of Application Functions

Arithmetic

The simplest example of refinement leading to specific requirement details is in the category of arithmetic functions. Arithmetic functions can be analyzed in terms of operands (the types of numbers to be dealt with) and operations (the desired types of numeric manipulation). Operands are characterized by precision, which is the number of bits required to maintain the complete information content of a data item (e.g., 12-14 bits for sensor data), and dynamic range, which is the working range needed between minimum and maximum values (e.g., distances from 1 to 10^6 yards (1 to 10^6 meters)). Operations involve manipulation of the types of data such as integer addition, subtraction, and questions such as whether high-precision floating point comparison operators or special support for trigonometric functions are required. All these kinds of requirements imply specific computer capabilities which, if neglected at the architectural level, can result in poor cost/performance designs.

Data Management

Data management is another key function. Sometimes there are large amounts of input data ("external" data) to be dealt with and special output formats to be generated, and almost always there is a whole range of internal data manipulation to be considered. Without architectural support, such manipulations can be cumbersome and extremely time-consuming. Appropriate addressing techniques (for words, bytes or bits, for example) and operations on various-sized units of data must also be considered.

Display

Display requirements, in general, can include special character processing, conversion routines, and a great deal of data structure manipulation during construction of display files. For example, the symbols used for the Naval Tactical Data System (NTDS) are not part of a commercial standard character set. Data communication uses similar functions and has additional protocol and error recovery considerations such as automatic parity generation. Both of these depend heavily on supporting functions which are closely allied to those of data management.

Armed with specific requirements such as those indicated above, one can analyze existing and proposed systems to see what needs are being satisfactorily addressed in contrast to what needs remain unmet. If the relative importance and frequency of usage of the particular features can be assessed, future systems can be designed with specific cost/performance architectural tradeoffs evaluated.

External devices that will be handled by computer systems may also have an impact on the architectural requirements of the computers. Some military devices have characteristics with direct commercial analogs (military radars and FAA radars; sonar sensors and seismic exploration sensors; military nuclear power plants and commercial power plants) while others (missiles versus manned space shots) are less comparable. The diversity of such equipment affects the computer instruction set architecture mostly in terms of data type requirements and is usually more applicable to determining the I/O architecture requirements.

Specific types of I/O related functions can be classified according to the kinds and amounts of data the interfaces handle and according to the functional sophistication implemented in the interfaces. Such data requirements have a definite impact on interface definitions and may influence internal architectures. I/O performance is also greatly affected by the interaction between the systems and their use by military personnel. System operator decision times and processing rates, as well as the frequency of use of various operational modes, have effects that are largely unknown and that are very rarely measured. Commercial systems are beginning to examine operator/system interactions through both measurement and modeling techniques. Their studies will provide valuable information that can be used with similar data gathered from military situations to improve the appropriate man/machine interfaces.

PROGRAM STATISTICS

Another way to characterize any application is to extrapolate from a statistical summary of its behavior on existing computers. This may be the best approach in the tactical data-processing case, where insubstantial information exists on specific architectural requirements, but where a variety of tactical programs have been implemented across a hardware/software spectrum. Analysis of these systems can provide some specific information (which may only reflect past requirements) concerning architectural details. It must be recognized that such details exhibit a certain degree of interdependence between the software implementations and the architectural features that were available on the particular computers programmed. Notice that in most cases these computers were allegedly designed to meet military requirements.

The IBM Federal Systems Division evaluates proposed and existing computer architectures by using benchmarks,[5] a technique also useful in characterizing applications. The same parameters (table 2) whose benchmarks are designed to measure across competing architectures are equally valid in establishing the salient features of existing systems for a given application area. The appropriate benchmarks, i.e., programs allowing evaluation of the parameters, can be programmed and run on existing computer hardware to see how well needs are currently met.

Table 2. Architecture Parameters

- instruction execution time
- addressing mode frequency and capability
- instruction format and field usage
- instruction set richness
- extensibility of architecture to accommodate special instructions
- register structure and usage
- interrupt handling facilities
- I/O facilities

These parameters have also been used to establish a set of specific architectural characteristics (table 3) for which performance statistics can be gathered. Such data, coupled with consideration of the interdependency between what is used and what is available, should help to establish the architectural requirements of the application of interest.

Table 3. Measurable Architectural Characteristics

1. For the most frequently executed and representative application programs:

   a. execution-time counts and percentage of occurrence of

      - each instruction type
      - register usage
      - various length instructions
      - indexed and nonindexed instructions
      - direct and indirect addressing instructions
      - data type usage (bit, character, half word, full word, double precision, floating point, etc.)
      - register operands and memory operands

   b. execution times of longest, shortest, and most common paths through each program, including execution time of each occurrence of each subroutine

Table 3. (Cont'd) Measurable Architectural Characteristics

c. numbers of significant address bits and percentage of occurrence of each needed for displacements on branches and on memory references

2. Context switching overheads (time, memory, and register requirements):

a. task to interrupt service routine

b. interrupt to interrupt service routine to task

c. task to task

Times may be in units of memory cycles. Overhead should include all between the last instruction executed (in interrupt service, for example) and the first instruction executed (in the user task).

3. Usage of intertask communication facilities--frequency and percentage of occurrence of:

a. shared data areas

b. global flags and pointers

c. message queuing

d. other

4. Arithmetic precision and dynamic range requirements:

a. input data

b. calculational

c. output

d. data formats (i.e., floating point, double precision, fixed, etc.) used for representation (frequency and percentage of occurrence)

5. Subroutine linkage conventions:

a. linkage overhead

- CALL (from caller to subroutine)
- RETURN (from subroutine to caller)

b. parameter passing--frequency and percentage of occurrence of

- method (e.g., by name, by value, etc.)
- number of parameters
- types of parameters

The most effective general method of gathering statistics on current architectures is to instrument operational systems with special hardware and software to watch for and record occurrences of appropriate events. Instrumentation can be as simple as a small digital counter selectively connected to count certain pulses, or as sophisticated as special operating system facilities to trap certain types of instructions or to time-stamp events. The cost of implementing any of these tools definitely depends as much on their timely introduction into system development plans as on the inherent complexity of the mechanisms. It is generally more expensive to add monitoring features to an existing system than it is to plan and develop them along with the system itself.

Current commercial data processing systems increasingly have built-in features for both online performance monitoring and resource control. Operating systems such as Univac/1108 Exec 8 and IBM OS/VS2 are typical. Few military tactical systems have recognized (with funding) the high payoff such data-gathering and performance facilities can provide. Typically, information is usually collected only by manufacturers (like Univac or IBM) for their own market research and consequently, almost no literature exists that characterizes current tactical systems using this approach.

ARCHITECTURAL IMPACT OF APPLICATION DATA

In spite of the value of facts and figures in establishing application characteristics that can be used to influence the designs and directions of new computer architectures, some commercial and many military design efforts proceed in a much less formal and ad hoc manner. Marketing considerations and user suggestions seem to have the major impact with little publicity allocated to either. A typical example of the current methodology used to design new computer architectures is given in a recent paper[6] titled "A Computer Architecture for an Advanced Real-Time Processing System" (ARPS). In giving design decisions, the authors relate that, "The design approach used with ARPS was to ascertain the problems our customers were encountering with their current machines. The result of this survey was a list of marketing and technical requirements that any real-time computer product should try to meet." The six major architectural requirements the ARPS was to satisfy were to:

1. Provide a virtual memory mechanism as an option

2. Provide stack facilities

3. Provide a system address space that was not limited by the instruction format

4. Use a tagged architecture concept for the CPU

5. Provide a fast context switch capability

6. Improve the I/O concepts being used on previous DOD systems.

Most of these requirements seem to stem from what is commercially available and "architecturally in vogue." The authors did not offer any evidence that these requirements are directly traceable to firm application requirements. One exception to this ad hoc design tendency was documented by IBM for the S/360.[7,8]

CHARACTERIZING TACTICAL MILITARY DATA PROCESSING

DATA PROCESSING VERSUS SIGNAL PROCESSING

It is necessary, in characterizing tactical military applications, to distinguish between data processing and special purpose applications such as signal processing. For example, it has been recognized[9] that signal processing architectural requirements can be factored into a few high data rate kernel operations (complex addition and multiplication, etc.) plus a control function that can be performed by a general purpose computer. Consequently, an appropriate architecture for signal processing is a special-purpose limited-operation high data rate computer coupled to a general-purpose control computer. This is in sharp contrast to a general-purpose computer that should be designed to efficiently handle a wide variety of data processing and resource management functions. In other words, greater flexibility in function and wider applicability is a highly desirable CPU characteristic. This consideration reinforces the concept of a cost/performance spectrum of implementations based on a standard family architecture, as detailed by the CFA program.

FUNCTIONAL CHARACTERIZATION

From the functional overview approach, current tactical data processing requires computers capable of managing the following (application-appropriate) resources:

- sensors
- communications
- weapons
- displays
- data.

Programs written for two military tactical computers (the AN/UYK-7 and the USQ-20) were used to gather characterization data for military application. Specifically, AN/UYK-7 fire control and sonar system programs under development were analyzed by gathering statistics from source program listings. This approach was dictated by the fact that

suitable computer system instrumentation did not exist and it was too costly to develop and install it for this study. Because frequency of source code occurrence and frequency of execution are not the same, the derived statistics will indicate trends in application characteristics rather than the actual values needed during program execution (table 4).

Table 4. Source Code Summary for a Tactical Data Processing System* (AN/UYK-7 CPU)

1. Instruction set utilization

48 out of 178 instructions accounted for 90% of code examined

50 out of 178 instructions were completely unused

2. Register utilization

	Set	Total	Percentage of total number of references	Number of registers to cover 50% of set references	Number of registers to cover 90% of set references
(arith)	A	8	51	2	6
(index)	B	8	27	3	7
(base)	S	8	22	2	4

3. Operand types (register, memory)

Module	Percentage of register	Register/memory ratio
1	42	.72
2	61	1.54
3	57	1.34
4	58	1.36
all	57	1.32

*Items 1-6 are from a NUSC study reported by Tom Conrad.[10]
Items 7-9 are from a NUSC study reported by Ed Hayes.[11]

Table 4. (Cont'd) Source Code Summary for a Tactical Data Processing System (AN/UYK-7 CPU)

4.	Use of indexed instructions		29% overall
5.	Use of indirect addressing		3.5% overall
6.	Mean of length of jumps		119 instruction lines overall (79% < 100)
7.	Average interrupt overhead		
	(all classes)		17.6 μs 23.5 cycles
8.	Context switching overhead		
	average interrupt handler to service routine (all classes)		4.5 μs 6 cycles
	task to task (average weighted by static entry frequency)		73 cycles
9.	Parameter passing		
	mean number of parameters passed (per subroutine call)		2.5
	transfer method: (percent)	accumulators	56
		index registers	2.6
		core memory	19
	parameter types: (percent)	integer	60
		real	4
		half circle*	33
		binary string	2

*Half circle is a bit string technique for representing angles.

Execution statistics were gathered from operational tactical programs on a USQ-20 computer. The USQ-20 is a single-accumulator CPU and, thus, a rather different architecture from multiple-register machines. The data presented in table 5 are the only available sample of execution statistics, and, thus, should not be compared with the source code statistics of the AN/UYK-7 programs.

Table 5. Execution Summary for a Tactical Data Processing System (USQ-20 Computer)

1. Instruction set utilization[12]

   Eleven out of 62 instructions accounted for 90% of code executed.

   Eight out of 62 instructions were completely unused.

2. Register utilization

   1 accumulator
   1 arithmetic extension register
   7 index registers

Reg #	Percentage of total number of references
0 (no indexing)	76
1	1
2	1
3	8
4	3
5	4
6	4
7	2

3. Operand types (recall only 1 accumulator and 1 extension):

   Less than 2% of instructions executed are strictly register reference (shifts).

4. Use of indexed instructions: 24% overall

5. Use of indirect addressing: not available

It is often stated that tactical military data processing has several unique characteristics that set it apart from commercial data processing. One of the first requirements usually put forth is realtime requirements or scheduling constraints. These have been discussed for command and control systems elsewhere,[13] and are summarized here:

> The most critical aspects include the time tolerances associated with the scheduling/dispatching and processing of tactical tasks, and the dynamic attribute of an ever changing, highly unpredictable tactical environment....These dynamic attributes of a tactical executive set it apart from the typical non-tactical executive, which operates in a well structured and fairly predictable environment, however it is obvious that many similarities do exist and in fact frequently outweigh the differences.

Particular examples that can be put forth as having time constraints equally stringent as military tactical systems are some of the NASA space and certain FAA applications. These examples also share another feature with tactical systems, that is, the "critical" or "life-and-death" nature of decisions (often considered unique to command/control systems). The analogy between tactical and nonmilitary does not, however, include translation of system failures into lost defense objectives.[14]

Tactical data base management is sometimes posed as different from any other kind of data management, mainly in terms of the requirement to establish data validity. During a tactical operation the data base may decay and build up simultaneously.[15] This means that just at the time a command decision may be necessary, valid and erroneous supporting data may be all mixed up together. Even after the operation it is no easy task to sort out and save only the correct information.

The preceding examples, whether viewed as unique features of tactical data processing or not, should be examined and refined to see what impact they have on computer architectural requirements. The ability to react to real-time environment-initiated tasks implies raw processing speed, speed that is derived basically from technological advances in switching speeds and from other implementation-dependent technology items (e.g., cache memory systems) that are invisible to a programmer at the instruction set level. Similarly, the ability to store and recall data quickly depends more on secondary storage access times and on channel data rates than on the available I/O instruction set. These unique features have no architectural impact at all beyond some general requirements that nearly every current commercially available computer satisfies in one form or another (such as interrupt processing systems for fast response to online devices, and disk systems for online data access).

It is also interesting to consider current military approaches to some of the other data processing characteristics where tactical demands are said to exceed commercial requirements such as reliability, availability, error control, and data security. Reliability can be provided in hardware by simple redundancy coupled with automatic self-checking procedures and by environmental hardening of the components. Certain important military applications have used such complete system redundancy because fault-tolerant hardware design is not mature enough to have been widely applied other than in some high-performance commercial machines and in selected space applications. It has been true that military hardware is generally more reliable and less susceptible to failure than commercial hardware. It cannot be assumed that this trend will continue because of certain market forces at work in the low end of commercial product lines. The wide application of minicomputers and microcomputers has encouraged environmental hardening of the computer components in order to gain access to an ever-expanding marketplace. It may well be that such commercial computer hardware will soon be sufficiently reliable for direct military use.

Software reliability is widely touted as necessary, but seldom planned for (in money or computer storage allocation), and frequently neglected in military systems. There is usually little enough memory planned for single copies of software modules, much less extra room for back-up, alternate, or checking copies. Software reliability does depend, to a large degree, on the architectural features of a machine. Errors in programming frequently result in attempts to execute operations that do not exist, and to store or retrieve data at locations that do not exist. In a well-defined machine (e.g., the more widely used commercial ones), these errors are detected and displayed so that they are easily handled by error recovery software, while in an ill-defined machine (e.g., AN/UYK-20) the results of such errors may be unpredictable or even undetectable. In general, commercial architectures are simply more well-defined than military ones.

Error checking and data validation are other aspects of military software that are sometimes neglected; therefore, even programs proven to be correct can go astray when given bad data. Yet experience has shown that the first thing cut from programs when size limitations are reached is the code that checks data against bounds or that provides for error recovery. This tends to be less true in the commercial world where market demands for reliability can be equal in scope to those for increased function.

Data security is currently an important commercial concern as well as a military one. Recent privacy regulations have begun to push manufacturers to provide facilities that should eventually be adaptable to military situations. The goal is to make the cost of getting access to

restricted information considerably higher than any value such information could have. It is to be expected that industry will concentrate a considerable research and development effort on architectures that will enhance data security. The military should track such work in order to capitalize on the appropriate results.

To summarize, nearly every characteristic that is usually put forth as a special requirement of military tactical data processing has already been addressed by commercial data processing with considerably more attention and detail. Therefore, although specific command/control application programs will still have to be written particularly for tactical systems (and will not be commercially available as off-the-shelf packages), the computer architectural features required do not differ from those also needed in commercial data processing applications.

ANALOGOUS COMMERCIAL DATA PROCESSING CHARACTERISTICS

A functional overview of commercial data processing is generally expressed as management of certain classes of resources such as processors, memory, peripheral devices and information. This characterization emphasizes functions that are common to all applications. Details are added, as appropriate, to build particular applications upon a common functional framework. It is this emphasis on commonality that makes the overview look different from that of tactical data processing, where application specifics intrude even at the higher functional levels.

An evaluation of a commercial computer architecture is summarized in table 6. Comparing these results with tables 4 and 5 for the AN/UYK-7 provides some interesting facts. Less than 30 percent of each instruction set is used to code 90 percent or more of the programs studied, whereas 25 and 35 percent of the instruction sets were not used at all. Register usage is higher in the UYK-7 than in the KA-10, where the KA-10 registers are general-purpose instead of being divided into special-usage sets. Without more detailed studies that set out to measure the same characteristics in the same way for both military and commercial systems, little else can be said in comparing the two types.

Table 6. A Summary of the Results from a Commercial System Study

DEC System 10 (KA-10 CPU)[16]

1. Instruction set utilization

128 out of 421 instructions would suffice for 98.8% of code examined.

147 out of 421 instructions were completely unused.

Table 6. (Cont'd) A Summary of the Results from a Commercial System Study

2. Register utilization (18 registers available)

Fewer than 10 were used for 90% of instructions counted in all 41 programs.

Fewer than eight were used for 98% of instructions counted in 29 out the 41 programs.

If only eight registers were available, the instruction counts would increase by only 20% for all 41 programs.

3. Calling sequences

Instruction counts for calling sequences can be as high as 25% of the total instruction count.

The breadth of application of various commercial computers contains an important lesson for the military tactical data processing community. A wide range of capabilities is now represented across the hardware spectrum of maxi-, mini-, and microcomputers. The commercial world has found it highly desirable to maintain compatibility across all these levels insofar as possible. Where it is not possible, they aim at least for common development tools so that customers can be convinced to choose equipment that is most appropriate from the hardware spectrum for particular applications. In this way, system growth requiring new hardware will have minimal impact on already developed software. Compatibility across hardware is then aimed at allowing software retention throughout system growth. Since military software development and maintenance represent a growing proportion of system costs, this approach should be of great interest.

Commercial application breadth can also be characterized in terms of a system spectrum. To look at a single example, IBM S/360 family of machines and software are in use over a wide range that includes the data base management of billions of bytes of data to the limited 65K-byte memory on the 16-inch (40 centimeters) space-qualified Hybrid Technology Computer (HTC) version from IBM Federal Systems Division.

One way to view the projected developments in commercial data processing is to consider economic impacts. The user software base for each of the major commercial lines has grown large enough to provide considerable economic inertia against any radical system changes. Thus it is to be expected that development will continue for some time to be evolutionary rather than revolutionary.

COMMERCIAL MACHINES IN TACTICAL USE

Commercial machines have many times been used successfully in various types of military tactical systems. A study of U.S. Army Electronics Command tactical systems showed that in March 1974, 26 projects had chosen 13 different computers (10 others had not yet specified a computer). Eleven of those 26 projects had, in fact, chosen commercial machines, ranging from an INTEL 8808 microprocessor to a ROLM 1602 minicomputer (a militarized NOVA) and a Burroughs B3500.[17]

The ROLM 1602 is a particular example of a machine directly derived from a standard commercial architecture and is implemented in military standard hardware for tactical use. The 1602 architecture is based on the NOVA series of compatible machines built by the Data General Corporation. It was ruggedized and repackaged to fit the standard air transport rack (ATR) form factor and qualified for the environmental specifications of MIL-E-5400, MIL-E-4158, and MIL-E-16400. By design then, the 1602 is a militarized machine capable of directly executing standard commercial NOVA code. At the time ROLM elected to work from the NOVA architecture, approximately 5000 NOVA machines were delivered with a considerable support software and peripheral base.

The Navy standard AN/UYK-20 minicomputer is also based on a commercially available architecture, the UNIVAC 1616. At the time of selection, however, only a couple hundred 1616 machines were delivered and very little support software existed. It was shown by MITRE,[18] in an excellent review of computer use, that large projects (such as the Defense Support program), which made extensive use of widely supported commercial computers and software, were able to control costs and schedules much better than those that used special-purpose processors with little support software.

A BRIEF HISTORY OF MILITARY DATA PROCESSING

The separation of commercial and tactical or military data processing equipment originated in the early 1960's. Available technology limited the implementation of a computer architecture to physically large systems whose volume was approximately several hundred cubic feet, whose power requirements were several thousand watts, and whose cost was several million dollars. This was reflected in an early DOD directive[19] and later amplified concerning the responsibilities of the Administration of the Automatic Data Processing Program when the definition section of the directive attempted to clarify Automatic Data Processing Equipment (ADPE) resources. Part of the definition section from DOD Directive 5100.40 of 18 May 1972 is included below:

Definitions

I. ADP Resources are defined as the totality of:

A. Automatic Data Processing Equipment (ADPE) -- General purpose, commercially available automatic data processing components and the equipment systems created from them, regardless of use, size, capacity, or price, which are designed to be applied to the solution or processing of a variety of problems or applications and which are not specifically designed, as opposed to configured, for any specific application.

1. This definition includes:

a. Digital, analog, or hybrid computer equipment;

b. Auxiliary or accessorial equipment such as data communications terminals, tape cleaners, tape testers, source data automation recording equipment (e.g., optical character recognition equipment, paper tape typewriters, magnetic tape cartridge typewriters, and other data acquisition devices), data output equipment (e.g., digital plotters, computer output microfilmers), etc., to be used in substantial support of digital, analog, or hybrid computer equipment, either cable-connected, wire-connected, or self-standing and whether selected or acquired with a computer, or separately; and

c. Punched card accounting machines (PCM) used in conjunction with or independently of digital, analog, or hybrid computers.

2. This definition excludes, except for reporting and as may be directed by the Secretary of Defense, computer equipment which is integral to a combat weapons system when:

a. It is physically incorporated into the weapon, or

b. It is integral to the weapons system from a design and procurement and operations viewpoint, or

c. Separate selection, acquisition, and/or management of the computer equipment would be infeasible.

> For the purpose of this Directive, being integral to a combat weapons system means being dedicated to and essential in real time to the performance of the mission of the weapons system in combat; e.g., automatic combat command, control and communications processing for specific combat weapons. Computer equipment used for logistic or administrative support of a weapons system, or which can be selected and acquired from commercial product lines independent of other components of the weapons system, is not covered by this exclusion. For purposes of this definition, a combat weapons system is an instrument of combat either offensive or defensive, used to destroy, injure, or threaten the enemy. It consists of the total entity that is an instrument of combat, which may incorporate in itself a complex assembly of functional parts, e.g., F-104 aircraft, FBM submarine, M-60 tank, Hawk missile.
>
> The purpose of the exclusions is to maintain in the program office the full responsibility for the RDT&E of the combat weapons system in which computers are subordinate elements.

Although the intent of the directive is to achieve standardization and economy in the use of DOD ADPE, it excludes the standardization of computers integral to weapon systems apparently because they are not viewed as general-purpose data processing machines but as specialized electronic components. This separation, of course, is favorable to defense contractors who could lock in their own computer hardware and software to a weapon system and the often necessary follow-on backfits and upgrades. To some extent this was alleviated in the Navy by the recognition of the need for a standard tactical data processor for weapon systems. However, the intrinsic notion of tactical data processing being different is still pursued and divorced from commercial ADPE.

One example of how the Navy has specified computer requirements in the past, is provided by the Navy's AN/UYK-20 procurement specification.[20] The architectural requirements as written can be satisfied by nearly any commercially available minicomputer. One detail that does have computer architectural impact is the requirement that negative zero never show up in any register accessible to the users. This implies use of two's-complement arithmetic or hardware trap and conversion of the negative zero in one's-complement arithmetic. The majority of commercial minicomputers use two's-complement arithmetic, automatically satisfying the requirement.

A different example of the lack of coordination between requirements and processor choice has come from an analysis of the A-6E aircraft navigation computer (by Professor Kodres of the Naval Postgraduate

School), which shows a mismatch between the precision of the input data and the computational precision maintained through the navigational calculations. An alternate microcomputer implementation has been studied to take advantage of the observed modularity in computer operations and of the low cost of microprocessor chips. Such a microcomputer system allowed cost-effective introduction of floating-point arithmetic in place of scaled fixed-point, making applications programming considerably more convenient.

CURRENT AND FUTURE REQUIREMENTS

The military has spent and is spending considerable effort attempting to define future computer and computer system requirements. A study for the Navy and Marine Corps[21] begins with functional requirements and refines them for specific details. Table 7 represents an extraction of those computer requirements having some kind of architectural impact. These requirements can again be satisfied by nearly every major commercial minicomputer.

Table 8 gives functional requirements detailed in an Air Force study of future data processing requirements.[14] It is noted that most of these are within today's software technology capabilities. It will be the special nature of command and control itself that will keep the necessary application software from becoming commercially available off-the-shelf.

Table 7. Future Navy Computer Architectural Requirements

- floating-point arithmetic
- fixed-point arithmetic
- conditional branching
- extensive interrupt system
- indirect addressing
  (min/max operations)
  (round-off operation)

Table 8. Future Air Force Computer Functional Requirements

- online data management and display
- computational assistance
- optimization (tactical)
- real-time data entry, reduction, and transmission
- online aids to command decisions
- real-time simulation exercise

The common factor among these studies is a lack of specific computer architectural details that imply the requirements for tactical military data processing are different from those of commercial data processing. What will be the most important issue is the balance to be achieved between the benefits of stabilizing the computer architecture and the desire to take advantage of possible growth in technology. It is extremely important not to lock out growth in attempting to provide standardization. Careful examination and determination of appropriate boundaries for specifying interface requirements will be necessary.

RECOMMENDATIONS

1. Support the Computer Family Architecture (CFA) program in implementing a baseline architecture for a military computer family.

2. Emphasize the availability of software development tools for the CFA baseline in order to reduce the cost of software development and maintenance.

3. Design instrumentation (software and hardware monitoring capabilities) into all current developing and future systems for analysis and evaluation.

4. Make sure all interface specifications and standardization developments are flexible enough to promote system incorporation of future technological advances.

5. Maintain a high level of awareness of commercial research and development in appropriate areas such as distributed processing and system reliability. Careful evaluation for architectural impact will be required, especially in terms of distributed processing and higher order languages (HOL's).

SUMMARY

In terms of computer architectural features required to support applications programs, tactical military data processing is the same as commercial data processing. It is, therefore, desirable economically to capitalize on available commercial technology in both hardware and software by choosing to develop a military computer family from a baseline commercial architecture. Care will be needed in interface specification and standardization development to allow for future growth. Limited military data processing budgets will require detailed cognizance of commercial investigations into research areas like higher order language (HOL) development and distributed processing techniques. Movement of applications programmers "away from the machine" through use of HOL's will allow any future computer architectural evolution to be implemented with minimal impact on the programmer and application program.

User community size and economic investment (usually in software) constitute a large force toward keeping computer architectural development evolutionary rather than revolutionary. Careful interface control will allow graceful incorporation of technological advances into existing systems. The same economic concerns will "push" commercial developments as well as military ones; the capability for the military to adapt commercial technology into their own systems will thus continue to be of major importance.

Many commercial and military computers are currently performing tactical military data processing tasks adequately although the support costs of maintaining a variety of unique systems is escalating. The question is whether a variety can continue to provide cost-effective solutions to military computing requirements in the future. If no is the accepted answer, it appears that standardization, which will take advantage of commercial technology, will offer a good alternate.

REFERENCES

1. "AADC Computer Family Architecture Request for Proposals," NRL 5490-112:WRS:vjs, Enclosure (1) to Naval Research Laboratory ltr 5490-117:WRS:vjs.

2. G. M. Amdahl, G. A. Blaauw, and F. P. Brooks, Jr., "Architecture of the IBM System/360," IBM Journal of Research and Development, April 1964, pp. 86-101.

3. G. C. Chapin, "What is Different About Tactical Military Operational Programs," AFIPS Conference Proceedings, Volume 42, 1973 National Computer Conference and Exposition, June 4-8, 1973, AFIPS Press, NJ, 1973.

4. E. C. Svendsen and D. L. Ream, "What's Different About the Hardware in Tactical Military Systems," AFIPS Conference Proceedings, Volume 42, 1973 National Computer Conference and Exposition, June 4-8, 1973, AFIPS Press, NJ, 1973.

5. R. L. Carberry, The Use of Benchmarks in the Evaluation of Computer Architectures, IBM Document 74-559-007, IBM Federal Systems Division, Electronics Systems Center, Owego, NY, 8 October 1974.

6. K. J. Thurber, J. C. Strauss, P. C. Patton, and R. C. Deward, "A Computer Architecture for an Advanced Real-Time Processing System," presented at COMPCON 76 East, 1976.

7. G. M. Amdahl, "The Structure of System/360. Part III - Processing Unit Design Considerations," IBM Systems Journal, vol. 3, no. 2, 1964, pp. 144-164.

8. G. M. Amdahl, G. A. Blaauw and F. P. Brooks, Jr., "Architecture of the IBM System/360," IBM Journal of Research and Development, vol. 8, no. 2, April 1964, pp. 87-101.

9. Y. S. Wu, Architectural Considerations of a Signal Processor Under Microprogrammed Control, Spring Joint Computer Conference, 1972.

10. T. Conrad, "Results of Fire Control Requirements Study," NUSC Memorandum No. SB122:TPC:ksa, Naval Underwater Systems Center, New London, CT, 1 December 1975.

11. E. Hayes, "AN/BQQ-5 Sonar Software Characterization," NUSC Memorandum Serial 325-253, Naval Underwater Systems Center, New London, CT, 9 August 1976.

REFERENCES (Cont'd)

12. Data was gathered at Fleet Combat Direction Systems Support Activity, San Diego, CA, by Steve Auguston.

13. W. G. Phillips, "What's Different About Tactical Executive Systems," AFIPS Conference Proceedings, Volume 42, 1973 National Computer Conference and Exposition, June 4-8, 1973, AFIPS Press, NJ, 1973.

14. D. W. Kosy, Air Force Command and Control Information Processing in the 1980's: Trends in Software Technology, Rand Corporation Report R-1012-PR, June 1974.

15. J. T. Dockery, "Remarks on Tactical Computer Problems," SIGPLAN Notices, vol. 10, no. 1, January 1975, pp. 176-177.

16. A. Lunde, Evaluation of Instruction Set Processor Architecture by Program Tracing, Department of Computer Science, Carnegie-Mellon University, PA, July 1974.

17. J. M. Apgar, F. S. Barricelli, A. H. Coleman, W. R. Light, Jr., and D. S. Morris, Survey of ECOM Tactical Computer-Based System ADP Requirements, Research and Development Technical Report ECOM-4328, Communications/ADP Laboratory, CENTACS, ECOM, Fort Monmouth, NJ, June 1975.

18. A. Asch, D. W. Kelliher, J. P. Locher III, and T. Connors, DOD Weapon Systems Software Acquisition and Management Study, Volume II, Supporting Material, MTR-6908, vol. II, Mitre Corporation, McLean, VA, June 1975.

19. DOD Directive 5100.40, 28 September 1963.

20. Naval Electronics Systems Command Contract Specification, Computer Digital Data, Combat System, ELEX-C-135, serial 288, 27 November 1972.

21. USN/USMC Future Data Systems Requirements and Their Impact on the AII Application Digital Computer (AADC), Computer Science Corporation, San Diego, CA.

BIBLIOGRAPHY

Advanced Weapons Management System (AWMS) Computer Requirements Report, Raytheon Company, Electromagnetic Systems Division, Goleta, CA, for Pacific Missile Center, Pt. Mugu, CA, 1 December 1975.

Alexander, B. L., All Applications Digital Computer (AADC) Application Studies for Advanced Light Attack Aircraft, Vought Aeronautics, LTV Aerospace Corporation, March 1973.

AN/BQS-13 DNA Sonar System Program Design Plan for Computer Executive Program for Use with AN/UYK-7 Computer, IBM Document No. 6934811R1, Contract No. N00024-70-C-1300, 17 May 1971.

Conrad, T. P., "Software Systems in Commercial and Tactical Computing Environments," NUSC Technical Memorandum No. SB122-4442-75, Naval Underwater Systems Center, New London, CT, 12 September 1975.

Cranston, D. L., "UYK-7 ASR SMD Measurement Report," NUSC Working Memorandum No. SB151-129-76, Naval Underwater Systems Center, New London, CT, 1 March 1976.

Culpepper, L. M., "A System for Reliable Engineering Software," IEEE Transactions on Software Engineering SE-1, vol. 2, June 1975, pp. 174-178.

DeMillo, R. A., "Part 1: General Language Considerations," Primitive for Tactical Real Time Control Languages Based on SIMULA 67, CENTACS Report No. 50, U. S. Army Electronics Command, Fort Monmouth, NJ, June 1975.

"DNA Computer and Software Analysis" (U), Design Data Package, AN/BQS-13 DNA, IBM Electronics Systems Center, Owego, NY, IBM Co. No. 3-260-7098R1, Contract No. N00024-70-C-1300 (CONFIDENTIAL).

Evaluation of the AADC Program of the United States Navy, Final Report of a Study by Research and Consulting, Inc., February 1975.

Goodenough, J. B., and L. H. Shafer, A Study of High Level Language Features, Softech, Inc., Waltham, MA, Research and Development Technical Report ECOM-75-0373-F, Volume 1, CENTACS, ECOM, Fort Monmouth, NJ.

Kilpatrick, P. S., W. C. Marshall, and E. A. Scales, All Semiconductor Distributed Aerospace Processor/Memory Study. Volume 1: Avionics Processing Requirements, Technical Report AFAL-TR-73-226, Honeywell, Inc. 1973.

BIBLIOGRAPHY (Cont'd)

Program Specification for Common Control Module MOD XI for Use with AN/UYK-7 Computer, NAVSHIPS 0967-029-5151, Contract No. N00024-70-C-1142, 1 June 1971.

Punj, D., S. E. Madnick, and J. D. DeTreville, A Survey of Navy Tactical Computer Applications and Executives by MIT, Family of Operating Systems No. 010-D, Sloan School of Management.

Report of the Operational Software Panel of the Navy Laboratory Computer Committee, September 1975.

Rubey, R. J., "What's Different about Tactical Military Languages and Compilers," AFIPS Conference Proceedings, Volume 42, 1973 National Computer Conference and Exposition, June 4-8, 1973, AFIPS Press, NJ, 1973.

Simulation of the AN/UYK-7 Computer Operations in the DNA System (U), IBM Electronics Systems Center, IBM CD 03-071-0029, 23 July 1971, Contract No. N00024-70-C-1300, (CONFIDENTIAL).

Slaughter, J. B., "Understanding the Software Problem," AFIPS Conference Proceedings Volume 43, 1974 National Computer Conference, AFIPS Press, NJ.

Solomon, M. B., Jr., "Economics of Scale and the IBM System/360," Communications of the ACM, vol. 9, no. 6, June 1966, pp. 435-440.

Tactical Computer Software Acquisition and Maintenance, Staff Study, 31 October 1973.

INITIAL DISTRIBUTION LIST

Addressee	No. of Copies
NAV SURFACE WEAPONS CENTER, WHITE OAK LABORATORY	1
NRL, Code 5400 (W. Smith)	1
NAVSEASYSCOM, SEA-032, -034, -06H1, -06H2, -615, -006 (W. H. Hill), PMS-393, -396	8
NAVAIRDEVCEN (C. Mattes)	1
NAVSURFWPNCEN (R. Hein)	1
NAVOCEANSYSCEN (J. K. Fogarty, N. L. Tinklepaugh)	1
NAVSEC, SEC-6178	1
NAVPGSCOL	1
NETC (F. Summer)	1
FCDSSA, SAN DIEGO (R. Estell)	1
PACMISRAN (P. Miller)	1
NATC (G. R. Ryan)	1
DDC, ALEXANDRIA	12